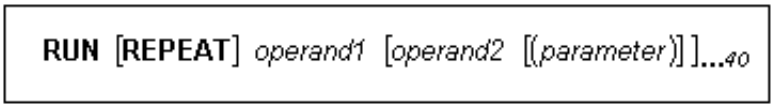
RUN



Operand	Possible Structure					Possible Formats											Referencing Permitted	Dynamic Definition
Operand1	C	S				A											yes	no
Operand2	C	S	A	G		A	N	P	I	F	B	D	T	L		G	yes	no

Function

The RUN statement is used to read a Natural source program from the Natural system file and then execute it.

REPEAT

RUN REPEAT causes the program not to prompt the user for input until the program has finished executing even if multiple output screens (produced by INPUT statements) are produced.

This feature may be used if the program is to display multiple screens of information without having the user respond to each screen.

Program Name - operand1

The name of the program can be specified as an alphanumeric constant or as the content of an alphanumeric variable. If a variable is used, it must be 8 characters in length.

The program may be stored in the current library or in a concatenated library (default steplib is SYSTEM). If the program is not found, an error message is issued.

The program is read into the source program work area and overlays any current source program.

Parameters - operand2

The RUN statement may also be used to pass parameters to the program to be run. A parameter may be defined with any format. The parameters are converted to a format suitable for a corresponding INPUT field. All parameters are placed on the top of the Natural stack.

The parameters can be read using an INPUT statement. The first INPUT statement issued will result in the insertion of all parameters into the fields specified in the INPUT statement. The INPUT statement must have the sign specification (SG=ON) for parameter fields defined with numeric format.

If more parameters are passed than are read by the next INPUT statement, the extra parameters are ignored. The number of parameters may be obtained with the system variable *DATA.

Note:
If operand2 is a time variable (format T), only the time component of the variable content is passed, but not the date component.

parameter

If *operand2* is a date variable, you can specify the session parameter DF as *parameter* for this variable. The session parameter DF is described in the Natural Reference documentation.

Dynamic Source Text Creation/Execution

The RUN statement may be used to dynamically compile and execute a program for which the source or parts thereof are created dynamically.

Dynamic source text creation is performed by placing source text into global variables and then referring to these variables by using "&" instead of "+" as the first character of the variable name in the source text. The content of the global variable will be interpreted as source text when the program is invoked using the RUN statement.

A global variable with index must not be used within a program that is invoked via a RUN statement.

It is not allowed to place a comment or an INCLUDE statement in a global variable.

Example

Program containing RUN statement:

```
/* EXAMPLE 'RUNEX1': RUN (WITH DYNAMIC SOURCE PROGRAM CREATION)
/************************************************************
/* GLOBAL AREA 'GDA1' CONTAINS '+CRITERIA'
/* +CRITERIA (A80)
/* PROGRAM CREATING SOURCE
DEFINE DATA GLOBAL USING GDA1
LOCAL
1 #NAME (A20)
1 #CITY (A20)
END-DEFINE
/************************************************************
INPUT 'Please specify the search values:' /
      'Name:' #NAME /
      'City:' #CITY
RESET +CRITERIA
/************************************************************
IF #NAME = ' ' AND #CITY = ' '
  REINPUT 'Enter at least 1 value'
END-IF
/************************************************************
IF #NAME NE ' '
  COMPRESS 'NAME' ' =''' #NAME '''' INTO +CRITERIA LEAVING NO
END-IF
IF #CITY NE ' '
  IF +CRITERIA NE ' '
    COMPRESS +CRITERIA 'AND' INTO +CRITERIA
  END-IF
  COMPRESS +CRITERIA ' CITY =''' #CITY '''' INTO +CRITERIA
END-IF
/************************************************************
RUN 'FIND-EMP'
/************************************************************
END
```

Program FIND-EMP executed by RUN statement:

```
/* PROGRAM EXECUTED WITH RUN STATEMENT ('FIND-EMP')
/************************************************
DEFINE DATA GLOBAL USING GDA1
 LOCAL
 1 EMPLOY-VIEW VIEW OF EMPLOYEES
 2 NAME
 2 CITY
END-DEFINE
/************************************************
FIND NUMBER EMPLOY-VIEW WITH &CRITERIA
                        RETAIN AS 'EMP-SET'
  DISPLAY *NUMBER
END
```

